**Βάσεις Δεδομένων 2017-2018**

**Εργασία 2η**

**Music Database**

**Ομάδα 42**

|  |  |  |
| --- | --- | --- |
| Σάββας Σαμπαζιώτης | 7974 | [sampazio@ece.auth.gr](mailto:sampazio@ece.auth.gr) |
| Γεωργόπουλος Κωνσταντίνος | 8031 | [georgopk@ece.auth.gr](mailto:georgopk@ece.auth.gr) |
| Ιωάννης Κόνιδας | 7308 | [konidas@ece.auth.gr](mailto:konidas@ece.auth.gr) |

**Εισαγωγή**

Το έγγραφο αυτό αποτελεί την αναφορά της 2ης εργασίας του μαθήματος Βάσεων Δεδομένων 2017-2018. Στο πρώτο μέρος, εμπεριέχεται λίστα με παραδείγματα όλων των περιορισμών τις βάσης, μαζί με τα σχετικά παραδείγματα. Στην δεύτερη ενότητα, εμπεριέχονται τα 10 αιτήματα μαζί τα σχετικά σχόλια και αποτελέσματα. Τέλος, Στην τρίτη και τελευταία ενότητα εμπεριέχονται σχόλια και παρατηρήσεις πάνω στην συνολική διαδικασία σχεδίασης της βάσης.

**Μέρος Α. Παραδείγματα Παραβίασης Περιορισμών.**

Στην ενότητα αυτή εμπεριέχεται μία λίστα με παραδείγματα αιτημάτων εισαγωγής για κάθε πίνακα της βάσης. Τα ερωτήματα αυτά είναι τέτοια ώστε να παραβιάζουν κάθε έναν από τους περιορισμούς των πινάκων αυτών.

|  |  |
| --- | --- |
| Πίνακας | 1.Store |
| Περιορισμός | * PRIMARY-KEY * Phone>1000000000 * Number >0 |
| Query | **INSERTINTO**`store`(`store\_name`,`phone`,`street`,`number`)  **VALUES** ('Null','2310999000','Karadjordjeva','10');  **INSERTINTO**`store`(`store\_name`,`phone`,`street`,`number`)  **VALUES** ('CD day','999000','Karadjordjeva','10');  **INSERTINTO**`store`(`store\_name`,`phone`,`street`,`number`)  **VALUES** ('CD day','2310999000','Karadjordjeva','-10'); |

|  |  |
| --- | --- |
| Πίνακας | 2.Is\_Available |
| Περιορισμός | * Not null (store\_name) * FOREIGN-KEY (store\_name) * Not null (street) * FOREIGN-KEY(street) * Not null (number) * FOREIGN-KEY(number) * Not null (album\_title) * FOREIGN-KEY(album\_title) * Not null (publisher) * FOREIGN-KEY (publisher) * Not null (quantity) * quantity ≥0 * Not null (price) * price≥0 |
| Query | **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**(**null**,**null**,**null**,**null**,**null**,**null**,**null**);  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('CD day','5','13241','Sheer Heart Attack','Sarm','Skra','4');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**(**null**,**null**,**null**,**null**,**null**,**null**,**null**);  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','5','13241','Sheer Heart Attack','Sarm','Karadjordjeva','4');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**(**null**,**null**,**null**,**null**,**null**,**null**,**null**);  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','5','13241','Sheer Heart Attack','Sarm','Skra','999');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**(**null**,**null**,**null**,**null**,**null**,**null**,**null**);  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','5','13241','boom','Sarm','Skra','4');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**(**null**,**null**,**null**,**null**,**null**,**null**,**null**);  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','5','13241','Sheer Heart Attack','CD Today','Skra','4');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**(**null**,**null**,**null**,**null**,**null**,**null**,**null**);  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','5','13241','Sheer Heart Attack','Sarm','Skra','4');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','-5','13241','Sheer Heart Attack','Sarm','Skra','4');  **INSERTINTO**`is\_available`(`store\_name`,`quantity`,`price`,`album\_title`,`publisher`,`street`,`number`)  **VALUES**('Stereo disc','5','13241','Sheer Heart Attack','Sarm','Skra','-4'); |

|  |  |
| --- | --- |
| Πίνακας | 3. Album |
| Περιορισμός | * Primary Key – Not Null(album\_title) |
| Query | **INSERT INTO** `album`(`album\_title`) **VALUES**(**NULL**);  **INSERT INTO** `album`(`publisher`) **VALUES**(**NULL**); |

|  |  |
| --- | --- |
| Πίνακας | 4. Included\_in |
| Περιορισμός | * Song\_order 1 * FOREIGN KEY (album\_title) * FOREIGN KEY (publisher) * FOREIGN KEY (song\_title) * Not NULL (song\_title) |
| Query | **INSERT** **INTO** `included\_in` (`song\_title`, `duration`) **VALUES** ('Bohemian Rhapsody', '-10');  **INSERT** **INTO** `included\_in` (`song\_title`, `duration`) **VALUES** (**NULL**, '15'); |

|  |  |
| --- | --- |
| Πίνακας | 5. Song |
| Περιορισμός | * Duration 0 * Primary Key – Not Null (song\_title) |
| Query | **INSERT** **INTO** `song` (`song\_title`, `duration`) **VALUES** ('unregistered song name', '-10');  **INSERT** **INTO** `song` (`song\_title`, `duration`) **VALUES** (**NULL**, '+10'); |

|  |  |
| --- | --- |
| Πίνακας | 6. Has\_Created |
| Περιορισμός | * FOREIGN KEY (artist\_name) * Not NULL (artist\_name) * FOREIGN KEY (song\_title) * Not NULL (song\_title) |
| Query | **INSERT** **INTO** `has\_created` (`artist\_name`, `song\_title`) **VALUES**( 'artist name that is not in the db', 'Bohemian Rhapsody');  **INSERT** **INTO** `has\_created` (`artist\_name`, `song\_title`) **VALUES**( **NULL**, 'Bohemian Rhapsody');  **INSERT** **INTO** `has\_created` (`artist\_name`, `song\_title`) **VALUES**( 'Queen', 'song that is not in the ');  **INSERT** **INTO** `has\_created` (`artist\_name`, `song\_title`) **VALUES**( 'Queen', **NULL**); |

|  |  |
| --- | --- |
| Πίνακας | 7. Artist |
| Περιορισμός | * Primary Key – Not Null (artist\_name) |
| Query | **INSERT** **INTO** `artist` (`artist\_name`) **VALUES** (**NULL**); |

|  |  |
| --- | --- |
| Πίνακας | 8. Musician |
| Περιορισμός | * FOREIGN KEY (artist\_name) * Not Null (artist\_name) |
| Query | **INSERT** **INTO** `musician` (`artist\_name`) **VALUES**( 'artist name that is not in the db');  **INSERT** **INTO** `musician` (`artist\_name`) **VALUES**( **NULL**); |
| Πίνακας | 9. Band |
| Περιορισμός | * FOREIGN-KEY (artist\_name) |
| Query | **INSERT INTO** `band`(`artist\_name`)**VALUES** ('John Konidas');  **INSERT INTO** `band`(`artist\_name`)**VALUES**(**NULL**); |

|  |  |
| --- | --- |
| Πίνακας | 10. Takes\_part\_in |
| Περιορισμός | * FOREIGN-KEY (band\_name) * FOREIGN-KEY (musician\_name) |
| Query | **INSERT INTO** `takes\_part\_in`(`band\_name`) **VALUES** ('savvassampaziotis');  **INSERT INTO** `takes\_part\_in`(`musician\_name`) **VALUES**('kostasgeorgopoulos');  **INSERT INTO** `takes\_part\_in`(`band\_name`) **VALUES**(**NULL**);  **INSERT INTO** `takes\_part\_in`(`musician\_name`) **VALUES**(**NULL**); |

|  |  |
| --- | --- |
| Πίνακας | 11. Genre |
| Περιορισμός | * FOREIGN-KEY (artist\_name) |
| Query | **INSERT INTO** `genre`(`artist\_name`)**VALUES**('Kostas konidas');  **INSERT INTO** `genre`(`artist\_name`)**VALUES**(**NULL**);  **INSERT INTO** `genre`(`genre`)**VALUES**(**NULL**); |

|  |  |
| --- | --- |
| Πίνακας | 12. Roles |
| Περιορισμός | * Not Null (role) * Not Null (musician\_name) * Not Null (band\_name) * FOREIGN KEY (musician\_name) * FOREIGN KEY (band\_name) |
| Query | **INSERT** **INTO** `role` (`musician\_name`, `band\_name`, `role`) **VALUES**  ('Freddie Mercury', 'Queen', **NULL**);  **INSERT** **INTO** `role` (`musician\_name`, `band\_name`, `role`) **VALUES**  (**NULL**, 'Queen', 'drummer');  **INSERT** **INTO** `role` (`musician\_name`, `band\_name`, `role`) **VALUES**  ('Freddie Mercury', **NULL**, 'drummer');  **INSERT** **INTO** `role` (`musician\_name`, `band\_name`, `role`) **VALUES**  ('Savvas Sampaziotis', 'Queen', 'drummer');  **INSERT** **INTO** `role` (`musician\_name`, `band\_name`, `role`) **VALUES**  ('Freddie Mercury', 'The Rock Cocks', 'drummer'); |

**Μέρος Β. Αιτήματα**

|  |  |
| --- | --- |
|  | Query No. 6 |
| Εντολή SQL | **SELECT** \* **FROM** song **WHERE** song.duration > (**SELECT** **AVG**(song.duration) **FROM** song); |
| Συμμετέχοντες  Πίνακες | * song { song\_title, duration, lyrics } |
| Περιγραφή | Η εντολή αυτή περιέχει εμφωλευμένο ερώτημα. Το εμφωλευμένο ερώτημα επιλέγει όλες τις τιμές του πεδίου *song.duration* και υπολογίζει τον μέσο όρο της διάρκειας των τραγουδιών. Το εξωτερικό ερώτημα, επιλέγει επίσης από τον πίνακα song όλα τα πεδία του, με όλες τις εγγραφές οι οποίες το γνώρισμα duration είναι μεγαλύτερο από τον μέσο όρο που υπολογίστηκε στο εμφωλευμένο. |
| Αποτελέσματα | | **song\_title** | **duration** | **lyrics** | | --- | --- | --- | | Bohemian Rapsody | 00:55:58 | Is this the real life? Is this just fantasy? | |

|  |  |
| --- | --- |
|  | Query No. 7 |
| Εντολή SQL | **SELECT** artist.artist\_name **FROM** artist **WHERE** artist.artist\_name  **NOT** **IN**  (**SELECT** **DISTINCT** has\_created.artist\_name **FROM** has\_created); |
| Συμμετέχοντες  Πίνακες | * artist { artist\_name, debut } * has\_created {artist\_name, song\_title} |
| Περιγραφή | Η εντολή αυτή πραγματοποιεί αφαίρεση συνόλων μεταξύ του των πινάκων artist και has\_created. Προκειμένου να είναι συμβατοί οι πίνακες, επιλέγεται το πεδίο artist\_name, το οποίο αποτελεί κλειδί και ξένο κλειδί αντίστοιχα σε κάθε πίνακα.  Σκοπός της εντολής είναι να προσδιορίσει τα ονόματα καλλιτεχνών οι οποίοι δεν έχουν καθόλου καταχωρημένα τραγούδια στην βάση.  Επειδή στην Maria DB 10.2 δεν υποστηρίζεται ο τελεστής EXCEPT, η υλοποίηση του τελεστή διαφοράς έγινε με τον τελεστή NOT IN |
| Αποτελέσματα | | **artist\_name** | | --- | | Blues Brothers | | Desperados | | Freddie Mercury | | Georgopoulos Anastasis | | John Lennon | | Los Mosquitos | | Mpokalidis Anastasis | | Skerlidis Sotiris | | Skerlidou Sofia | | Soski | | Tsiolakoglou Nikos | | Walk off the Earth | |

|  |  |
| --- | --- |
|  | Query No. 8 |
| Εντολή SQL | **SELECT** has\_created.song\_title,  included\_in.album\_title,  has\_created.artist\_name  **FROM** has\_created **INNER** **JOIN** included\_in **ON** included\_in.song\_title=has\_created.song\_title; |
| Συμμετέχοντες  Πίνακες | * included\_in {album\_title, publisher, song\_title, song\_order} * has\_created {artist\_name, song\_title} |
| Περιγραφή | Το ερώτημα αυτό υλοποιεί τον τελεστή τομής μεταξύ των πινάκων has\_created και included\_in.  Καθώς στην έκδοση 10.2 της Maria DB δεν υποστηρίζεται ο τελεστής INTERSECT, η τομή υλοποιήθηκε με INNER JOIN, πάνω στο ξένο κλειδί “song\_title”.  Ο σκοπός του ερωτήματος είναι να προσδιορίσει τις εγγραφές τραγουδιών (βάσει του κλειδιού “song\_title”) οι οποίες είναι καταχωρημένες και στους δύο πίνακες.  Το πεδίο “has\_created.artist\_name” επιλέγεται «ως εξτρά» και δεν έχει κάποια αξία στην διαδικασία τομής. |
| Αποτελέσματα | | **song\_title** | **album\_title** | **artist\_name** | | --- | --- | --- | | We Will Rock You | News of the World | Queen | | We Are the Champions | News of the World | Queen | | Baby Appeal | The Red Hot Chilli Peppers | Red Hot Chili Peppers | | Buckle Down | The Red Hot Chilli Peppers | Red Hot Chili Peppers | |

|  |  |
| --- | --- |
|  | Query No. 9 |
| Εντολή SQL | **SELECT** artist.artist\_name **FROM** artist **WHERE** artist.artist\_name  **IN**  (**SELECT** **DISTINCT** has\_created.artist\_name **FROM** has\_created); |
| Συμμετέχοντες  Πίνακες | * artist { artist\_name, debut } * has\_created {artist\_name, song\_title} |
| Περιγραφή | Το ερώτημα αυτό προκύπτει να είναι παρόμοιο με το Νο.7.  Σκοπός του ερωτήματος είναι τυπώσει τα ονόματα των καλλιτεχνών οι οποίοι έχουν τραγούδια καταχωρημένα στο όνομά τους στη βάση. |
| Αποτελέσματα | | **artist\_name** | | --- | | Queen | | Red Hot Chili Peppers | |

|  |  |
| --- | --- |
|  | Query No. 10 |
| Εντολή SQL | **CREATE** **OR** **REPLACE** **VIEW** album\_duration  **AS** **SELECT** artist.debut, band.logo, band.founding\_date, band.split\_date  **FROM** artist  **NATURAL** **JOIN** band  **ORDER** **BY** debut;  **SELECT** \* **FROM** album\_duration; |
| Συμμετέχοντες  Πίνακες | * artist {artist\_name, debut} * band {artist\_name, logo, founding\_date, split\_date} |
| Περιγραφή | Το ερώτημα αυτό αποτελείτε από 2 ξεχωριστά ερωτήματα.  Το πρώτο ερώτημα δημιουργεί μία όψη πίνακα με το όνομα “band\_info”. Η όψη αυτή επιλέγει όλα τα πεδία των πινάκων band και artist. Οι εικονικές καταγραφές της όψης αυτής αποτελείται από την φυσική συνένωση των δύο πινάκων πάνω στο κοινό κλειδί τους “artist\_name”  Το δεύτερο ερώτημα απλά επιλέγει όλα τα πεδία με όλες τις εγγραφές του εικονικού αυτού πίνακα. |
| Αποτελέσματα | | **debut** | **logo** | **founding\_date** | **split\_date** | | --- | --- | --- | --- | | 1970-00-00 | 0x436F6E6… | 1970-00-00 |  | | 1976-00-00 |  | 1976-00-00 |  | | 1983-00-00 | 0x43006F0… | 1983-00-00 |  | | 2006-00-00 | 0x35003… | 2006-00-00 |  | | 2010-00-00 |  | 2010-00-00 | 2012-00-00 | | 2012-00-00 |  | 2012-00-00 | 2014-00-00 | | 2014-00-00 |  | 2014-00-00 |  | |

**Μέρος Γ. Παρατηρήσεις**